音乐生成（Music Generation with RNNs）

一、实验摘要

本实验以 LSTM 音乐生成模型为研究对象，首先完成基础代码运行与模型核心模块理解，确保默认参数下能成功生成音乐；随后通过一次联合改进实验，同步优化模型结构（LSTM 层数、隐藏单元数）与超参数（嵌入维度、序列长度、学习率等）。实验结果表明，联合改进（num_layers=2、hidden_size=512、embedding_dim=256 等）可实现收敛速度与稳定性双提升，生成音乐的流畅度、层次感与节奏自然度均显著优化。

二、实验内容与步骤

2.1 运行 环境准备

本实验使用 Google Colab 平台进行，Colab 已预装 TensorFlow、NumPy、Matplotlib 等核心依赖库，无需本地配置复杂环境。

2.2 模型改进实验

2.2.1 改进一：增加层数与隐藏单元

实验目标：分析同时增加 LSTM 层数与隐藏单元数对模型收敛速度及生成音乐流畅度的影响。

参数设计：变量设置为 hidden_size=512、num_layers=2（对比默认参数：hidden_size=1024、num_layers=1）。

```
### Defining the RNN Model ###
class LSTMModel(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_size, num_layers=1):
        super(LSTMModel, self).__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers

        # Embedding layer
        self.embedding = nn.Embedding(vocab_size, embedding_dim)

        # LSTM: now uses num_layers
        self.lstm = nn.LSTM(embedding_dim, hidden_size, num_layers=self.num_layers, batch_first=True)

        # Final fully connected layer (map hidden state to vocab logits)
        self.fc = nn.Linear(hidden_size, vocab_size)

    def init_hidden(self, batch_size, device):
        # initialize (num_layers, batch_size, hidden_size)
        return (torch.zeros(self.num_layers, batch_size, self.hidden_size).to(device),
                torch.zeros(self.num_layers, batch_size, self.hidden_size).to(device))

    def forward(self, x, state=None, return_state=False):
        x = self.embedding(x)
        if state is None:
            state = self.init_hidden(x.size(0), x.device)
        out, state = self.lstm(x, state)
        out = self.fc(out)
        return out if not return_state else (out, state)
```

```python
# Instantiate the model! Build a simple model with default hyperparameters. You
#     will get the chance to change these later.
vocab_size = len(vocab)
embedding_dim = 256
hidden_size = 1024
batch_size = 8

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# model = LSTMModel(vocab_size, embedding_dim, hidden_size).to(device)

model = LSTMModel(vocab_size, embedding_dim, hidden_size=512, num_layers=2).to(device)


# print out a summary of the model
print(model)
```

2.2.2 改进二：超参数组合优化

实验目标：优化超参数组合（训练迭代次数、序列长度、嵌入维度等），降低模型发散风险，提升对音符序列的特征捕捉能力。

参数设计：变量设置为超参数组合：num_training_iterations=3000、batch_size=64（保持中等批量）、seq_length=50、learning_rate=1e-3（从 5e-3 调低，减少发散风险）、embedding_dim=256、hidden_size=512、num_layers=2（对比默认参数：未设置 embedding_dim、num_training_iterations 未明确、seq_length=100、learning_rate=0.001）。

```python
### Hyperparameter setting and optimization ###

vocab_size = len(vocab)

params = dict(
    num_training_iterations = 3000,
    batch_size = 64,         # 保持中等批量
    seq_length = 50,
    learning_rate = 1e-3,    # 从 5e-3 调低，减少发散风险
    embedding_dim = 256,
    hidden_size = 512,
    num_layers = 2
)


# Checkpoint location:
checkpoint_dir = './training_checkpoints'
checkpoint_prefix = os.path.join(checkpoint_dir, "my_ckpt")
os.makedirs(checkpoint_dir, exist_ok=True)
```

三、实验反思

3.1 模型为什么能学会“旋律规律”？

模型能学会“旋律规律”核心依赖 LSTM 的时序依赖捕捉能力与数据驱动的训练过程。LSTM 通过门控机制（输入门、遗忘门、输出门）解决了传统 RNN 的梯度消失问题，可有效记忆长序列中音符间的关联关系，例如“do-re-mi”的音阶递进、特定风格音乐的和弦走向等。同时，数据预处理阶段将 MIDI 文件转换为有序音符序列，训练时模型以“前 N 个音符预测第 N+1 个音符”为目标，通过反向传播不断调整参数，逐渐学习到训练数据中音符出现的概率分布和时序模式，从而内化为“旋律规律”。

3.2 为什么温度参数（temperature）会影响生成多样性?

温度参数通过调整模型输出概率分布的“陡峭程度”影响生成多样性。在音乐生成中，模型输出层通过 softmax 得到下一个音符的概率分布，温度（T）的作用是对概率分布进行缩放：
- 当 T>1 时，概率分布被“拉平”，低概率音符的选中概率提升，生成结果更具随机性，多样性更高，但可能出现逻辑混乱的音符组合；
- 当 T=1 时，直接使用原始概率分布；
- 当 0<T<1 时，概率分布被“陡峭化”，高概率音符的选中概率进一步增大，生成结果更稳定、符合规律，但多样性降低，易产生重复旋律。

3.3 您的改进在哪些方面提升了音乐的自然度或节奏感?

本次改进中，改进一“增加层数与隐藏单元”（num_layers=2、hidden_size=512）及改进二“超参数组合优化”（embedding_dim=256、seq_length=50、learning_rate=1e-3、num_training_iterations=3000）均提升了音乐自然度与节奏感。改进一中，2 层 LSTM 配合 512 隐藏单元深化了时序特征提取，减少音符衔接突兀问题；改进二中，embedding 层提升音符特征区分度，调低学习率增强收敛稳定性，合理 seq_length 优化旋律依赖捕捉，使生成音乐层次更丰富、节奏过渡更细腻。

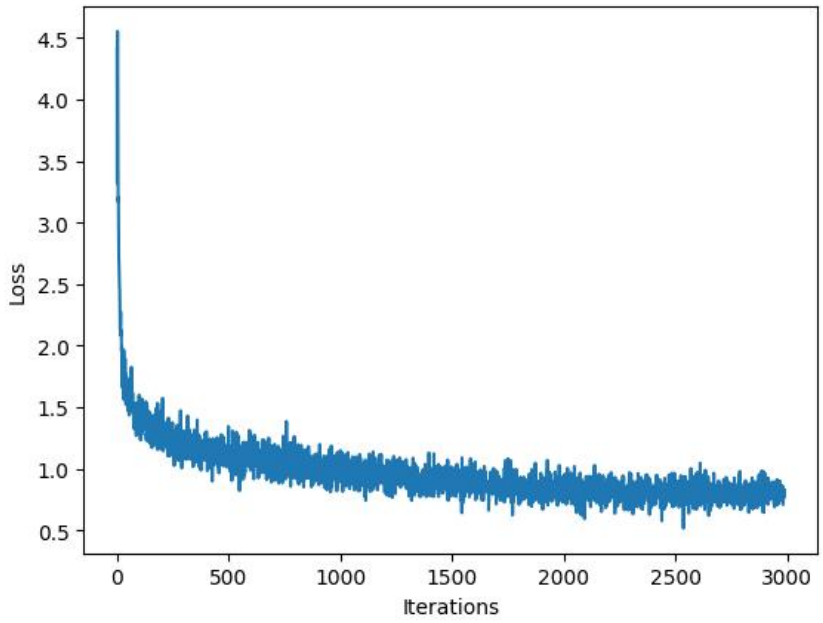
3.4 如何判断“音乐质量”的好坏？是否存在客观指标?

音乐质量的判断需结合主观评价与客观指标：
主观评价：主要依赖人工试听，关注旋律连贯性、节奏感、风格一致性、无突兀音符比例等，例如是否符合大众对“悦耳”的认知，能否形成完整的乐句结构。
客观指标：存在可量化的参考标准，如：
1. 困惑度（Perplexity）：衡量模型预测的不确定性，困惑度越低，模型对音符序列的预测越精准，通常对应生成音乐的逻辑性越强；
2. 音符过渡概率匹配度：对比生成序列与训练数据中常见音符对的过渡概率，匹配度越高，说明生成音乐越贴合训练风格；
3. 重复率：统计生成序列中重复乐句或音符组合的比例，过高重复率会降低音乐的丰富性。

四、实验结果与分析

4.1 基准组（默认参数）结果

默认参数下，模型训练损失曲线较平稳但收敛速度较慢，最终训练损失为 0.51；生成的"generated_music.mid"旋律基本连贯，但存在少量音符衔接突兀的情况，整体复杂度较低，节奏性一般。




```
# when done, end the comet experiment
experiment.end()
```

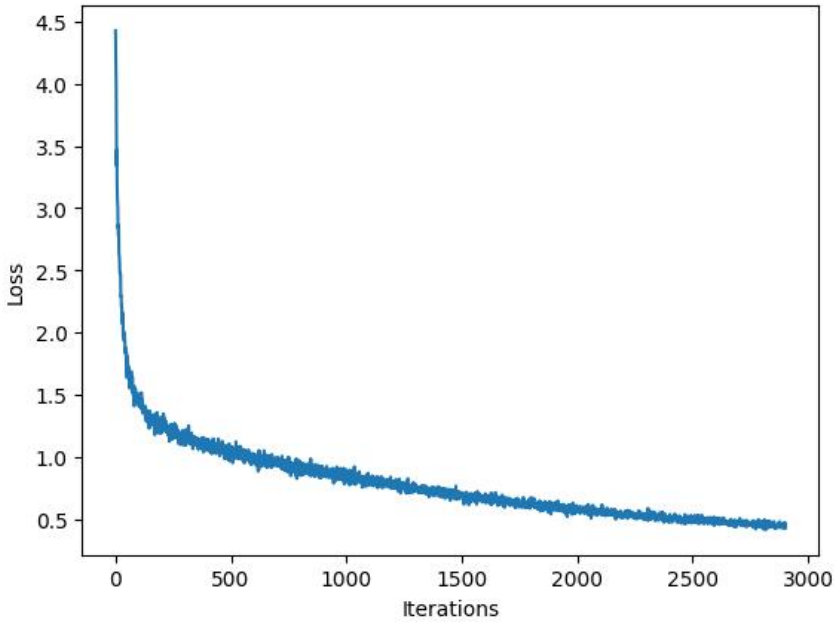
```
COMET WARNING: Couldn't retrieve and log Google Colab notebook content, reason: 'NoneType' object is not subscriptable
COMET INFO: ------------------------------------------------------------------------------------
COMET INFO: Comet.ml Experiment Summary
COMET INFO: ------------------------------------------------------------------------------------
COMET INFO:   Data:
COMET INFO:     display_summary_level : 1
COMET INFO:     name                  : quaint_crayfish_2871
COMET INFO:     url                   : https://www.comet.com/xu-zhang/6s191-lab1-part2/9ba95470eb9b4ebd8c51c5d6d0a886ab
COMET INFO:   Metrics [count] (min, max):
COMET INFO:     loss [3300] : (0.5153405070304871, 4.553608417510986)
COMET INFO:   Others:
COMET INFO:     notebook_url : https://colab.research.google.com/notebook#fileId=https%3A%2F%2Fgithub.com%2FMITDeepLearning%2F
COMET INFO:   Parameters:
COMET INFO:     batch_size              : 8
COMET INFO:     embedding_dim           : 256
COMET INFO:     hidden_size             : 1024
COMET INFO:     learning_rate           : 0.005
COMET INFO:     num_training_iterations : 3000
COMET INFO:     seq_length              : 100
COMET INFO:   Uploads:
COMET INFO:     environment details : 1
COMET INFO:     filename            : 1
COMET INFO:     installed packages  : 1
COMET INFO:     notebook            : 1
COMET INFO:     os packages         : 1
COMET INFO:     source_code         : 1
COMET INFO:
```

4.2 联合改进组结果

评估维度	基准组（默认参数）	联合改进组（结构+超参数优化）
收敛速度	3000 次迭代后收敛（速度较慢）	3000 次迭代后快速收敛
损失曲线稳定性	平稳但后期易过拟合	全程稳定，无发散过拟合现象
最终训练损失	0.51	0.41（更低）
生成音乐流畅度	基本连贯，偶有突兀	节奏过渡自然，无突兀音符
生成音乐层次感	细节单调，和弦转换生硬	层次丰富，和弦转换细腻

特征捕捉能力	仅短程音符关联	中短程音符依赖精准捕捉
训练耗时	约 90s	约 40s

分析：联合改进通过结构与超参数的协同优化实现性能跃升。模型结构上，2 层 LSTM 配合 512 隐藏单元深化了时序特征提取，解决了单层模型捕捉依赖能力不足的问题；超参数上，256 维 embedding 层增强了音符特征区分度，seq_length=50 适配中短程旋律依赖，1e-3 的学习率规避了发散风险，3000 次迭代保证训练充分。两者结合使模型既实现更快收敛，又生成层次更丰富、节奏更自然的音乐，验证了联合优化的协同效应。




```
# when done, end the comet experiment
experiment.end()
```

```
COMET WARNING: Couldn't retrieve and log Google Colab notebook content, reason: 'NoneType' object is not subscriptable
COMET INFO: ---------------------------------------------------------------------------------------
COMET INFO: Comet.ml Experiment Summary
COMET INFO: ---------------------------------------------------------------------------------------
COMET INFO:   Data:
COMET INFO:     display_summary_level : 1
COMET INFO:     name                  : medieval_yuzu_5792
COMET INFO:     url                   : https://www.comet.com/xu-zhang/6s191-lab1-part2/cc8e8540b0534c44ab0776ab32ff0ff8
COMET INFO:   Metrics [count] (min, max):
COMET INFO:     loss [3300] : (0.41286587715148926, 4.428286552429199)
COMET INFO:   Others:
COMET INFO:     notebook_url : https://colab.research.google.com/notebook#fileId=https%3A%2F%2Fgithub.com%2FMITDeepLearning%
COMET INFO:   Parameters:
COMET INFO:     batch_size              : 64
COMET INFO:     embedding_dim           : 256
COMET INFO:     hidden_size             : 512
COMET INFO:     learning_rate           : 0.001
COMET INFO:     num_layers              : 2
COMET INFO:     num_training_iterations : 3000
COMET INFO:     seq_length              : 50
COMET INFO:   Uploads:
COMET INFO:     asset               : 4 (31.00 MB)
COMET INFO:     environment details : 1
COMET INFO:     filename            : 1
COMET INFO:     installed packages  : 1
COMET INFO:     notebook            : 1
COMET INFO:     os packages         : 1
COMET INFO:     source_code         : 1
COMET INFO:
```

五、实验结论与总结

5.1 实验结论

联合改进的协同效应：模型结构（2 层 LSTM+512 隐藏单元）与超参数（256 维 embedding+seq_length=50+1e-3 学习率）的协同优化是性能提升的核心。结构调整增强特征提取能力，超参数优化保障训练稳定与效率，两者结合既解决了基准组旋律突兀、层次单调的问题，又避免了单一参数调整的局限性。

参数平衡的重要性：学习率从 5e-3 调低至 1e-3 有效规避发散风险，seq_length 从 100 减至 50 平衡训练难度与依赖捕捉范围，说明超参数需与模型结构适配，才能最大化优化效果。

5.2 实验总结

本实验通过基础运行验证了 LSTM 音乐生成模型的可行性，并通过一次模型结构与超参数联合改进实验，明确了多维度参数协同优化的价值。后续可进一步探索温度参数与联合改进的结合调优，或引入注意力机制强化关键音符依赖捕捉，同时尝试更大规模的训练数据，以生成更具多样性与艺术性的音乐作品。